



International Conference on Computational Science, ICCS 2013

## Dynamic data driven sensor array fusion for target detection and classification

Nurali Virani<sup>a</sup>, Shane Marcks<sup>a,b</sup>, Soumalya Sarkar<sup>a</sup>, Kushal Mukherjee<sup>a</sup>, Asok Ray<sup>a</sup>,  
Shashi Phoha<sup>b,\*</sup>

<sup>a</sup>Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, University Park, PA 16802, USA

<sup>b</sup>Applied Research Laboratory, The Pennsylvania State University, University Park, PA 16802, USA

---

### Abstract

Target detection and classification using unattended ground sensors (UGS) has been addressed in literature. Various techniques have been proposed for target detection, but target classification is a challenging task to accomplish using the limited processing power on each sensor module. The major hindrance in using these sensors reliably is, that, the sensor observations are significantly affected by external conditions, which are referred to as context. When the context is slowly time-varying (e.g., day-night cycling and seasonal variations) the usage of the same classifier may not be a good way to perform target classification. In this paper, a new framework is proposed as a Dynamic Data Driven Application System (DDDAS) to dynamically extract and use the knowledge of context as feedback in order to adaptively choose the appropriate classifiers and thereby enhance the target classification performance. The features are extracted by symbolic dynamic filtering (SDF) from the time series of sensors in an array and spatio-temporal aggregation of these features represents the context. Then, a context evolution model is constructed as a deterministic finite state automata (DFSA) and, for every context state in this DFSA, an event classifier is trained to classify the targets. The proposed technique of detection and classification has been compared with a traditional method of training classifiers without using any contextual information.

**Keywords:** Context Modeling; Symbolic Dynamics; Sensor fusion; DDDAS; Pattern Recognition

---

### 1. Introduction

Unattended ground sensors (UGS) have been used for event detection and classification in diverse applications, such as border security, military operations, and industrial monitoring, where a wide range of sensing modalities (e.g., acoustic, seismic, passive infrared, magnetic and piezoelectric) have been implemented. The UGS systems are comprised of compact sensing modules that usually have limited processing capabilities and may also be required to communicate the results of event detection and classification to a remote processing center for data logging and making decisions on target tracking. In this setting, Iyengar et al. [1] and Bland [2] have reported feasible techniques for event detection by making use of analytical tools like correlation analysis, copula, and linear predictive modeling. However, classification of event types is often limited due to occurrence of high false alarm rates, possibly because the on-board data processing algorithms are inadequate for the purpose at hand.

One of the major impediments in using UGS systems is that the sensors like seismic and acoustic, which have good potential to be used for personnel detection [3], are severely affected by daily and seasonal meteorological

---

\*Corresponding author. Tel: +1-814-863-8005; Email: [sxp26@arl.psu.edu](mailto:sxp26@arl.psu.edu)

variations [4]. Attempts have been made to analytically represent the effects of such meteorological variations on the sensors by using approximate environmental models [5]. Since a large number of sensor modules are deployed in a region, it might not be feasible to obtain all the time-varying environmental parameters (e.g., temperature, wind velocity, soil permeability, soil stiffness, and humidity) for every sensor location. In order to capture the effects of external conditions, a DDDAS is proposed as a feasible alternative to the model-based system. From the large ensemble of time series data available from the sensor arrays, a sense of “context” needs to be inferred dynamically as a representation of the changing environmental conditions and this information can be used to adapt the individual sensing modules by changing classifiers to improve the classification. In the current state of the art, a single set of classifier is often designed for all contexts, which may be a source of false alarm rates for a given probability of successful detection. This issue is addressed in this paper that proposes a procedure to replace the notion of a single set of classifiers with multiple sets of context-dependent classifiers, where the contexts are identified from spatially distributed sources of time series information.

The role of context in pattern recognition has been explored by many investigators and are extensively reported in literature [6]. Recently, efforts have been expended to build context awareness in sensor networks to estimate missing sensor values and identify faulty sensors. For example, Elnahrawy and Nath [7] have defined context in terms of the values of neighboring sensing nodes and the sensor’s own history, which is used to predict the next sensor reading. In the work reported in the current paper, a context is defined in terms of the external conditions that influence the output of sensors placed in a given region. For e.g., change in soil stiffness by precipitation, affecting output of seismic sensor, strong wind gusts affecting the output of acoustic sensors, etc.

Zhao et al. [8] have reported a framework for information aggregation for monitoring sensor networks with the objective of identifying system failures, performance degradation, and resource depletion. They mentioned that, in certain applications, the nodes might be able to autonomously tailor their performance based on the knowledge of aggregated properties. In contrast, the current paper proposes a sensor fusion technique that aggregates features from all the sensing nodes of the sensor array in the specified region to obtain a meta-data representation of the sensor network’s regional behavior. The aggregated feature is used to construct a symbolic model of the context to capture the changes in the network behavior on a relatively slow time scale, due to external conditions affecting the sensor data. Then, this context model is used to improve target detection and classification.

The major contributions of this paper are summarized as follows:

1. Development of an information aggregation technique to identify the context of the network’s regional behavior based on the data from a sensor array.
2. Construction of a symbolic context evolution model (having a deterministic finite state automaton (DFSA) structure) from the time series data of a sensor array.
3. Enhancement of the target detection and classification capability of each sensor by making use of the information about the context in the sensor network.
4. Validation of the test results for target detection and classification with and without considering context.

The paper is organized in five sections including the current section that highlights the objective, motivation and relevant literature. Section 2 and Section 3 have together delineated the proposed framework with the relevant theory. Section 4 summarizes the experiments and results. Section 5 points out the salient features of the proposed framework and provides directions for further research.

## 2. Problem Formulation and Methodology

This section first presents the mathematical preliminaries including pertinent definitions and their implications.

**Definition 2.1** (DFSA) A deterministic finite state automaton (DFSA) is a 3-tuple  $G = (\Sigma, Q, \delta)$  where:

- $\Sigma$  is a non-empty finite set, called the symbol alphabet, with cardinality  $|\Sigma| < \infty$ ;
- $Q$  is a non-empty finite set, called the set of states, with cardinality  $|Q| < \infty$ ;
- $\delta : Q \times \Sigma \rightarrow Q$  is the state transition map;

and  $\Sigma^*$  is the collection of all finite-length strings with symbols from the alphabet  $\Sigma$  and the (zero-length) empty string  $\epsilon$ .

**Definition 2.2** (PFSA) A probabilistic finite state automaton (PFSA) is constructed upon a DFSA  $G = (\Sigma, Q, \delta)$  as a pair  $K = (G, \pi)$ , i.e., the PFSA  $K$  is a 4-tuple  $K = (\Sigma, Q, \delta, \pi)$ , where:

- $\Sigma, Q$ , and  $\delta$  are the same as in Definition 2.1;
- $\pi : Q \times \Sigma \rightarrow [0, 1]$  is the probability morph function that satisfies the condition  $\sum_{\sigma \in \Sigma} \pi(q, \sigma) = 1 \quad \forall q \in Q$ , and  $\pi_{ij}$  is the probability of occurrence of a symbol  $\sigma_j \in \Sigma$  at the state  $q_i \in Q$ .

**Definition 2.3** (D-Markov [9][10]) A D-Markov machine is a PFSA in which each state is represented by a finite history of  $D$  symbols as defined by:

- $D$  is the depth of the Markov machine;
- $Q$  is the finite set of states with cardinality  $|Q| \leq |\Sigma|^D$ , i.e., the states are represented by equivalence classes of symbol strings of maximum length  $D$  where each symbol belongs to the alphabet  $\Sigma$ ;

**Remark 2.1** For the special case of Defn. 2.3, if  $|Q| = |\Sigma|^D$ , then the following condition holds: There exist  $a, b \in \Sigma$  and  $\sigma \in \Sigma^{D-1}$  such that  $\delta(a\sigma, b) = \sigma b$  and  $a\sigma, \sigma b \in Q$ .

### 2.1. Background

As explained in [11] as well as in Section 3, symbolic dynamic filtering (SDF) [9] has been used to represent the time series data as a PFSA. The stationary state probability vector of the PFSA, generated from a time series, is used as a feature for target detection and classification. A single set of classifiers, derived from the training data, may yield satisfactory results if the training data and test data are collected in similar conditions or if the symbolic analysis is robust enough to compensate for any signal variations due to disturbances [3]. However, for a more general scenario where the signal is affected significantly by external circumstances, referred as context in this paper, a single set of classifiers might become inadequate. A new framework which allows selection of a set of classifiers based on the context is developed in the symbolic framework as an extension to the existing framework.

### 2.2. Scenario

A generic scenario is presented in Fig. 1 to explain the implementation of the proposed framework. Let a sensor array be laid out in a geographical area (e.g., an array of sensor modules used for border security). This array is composed of several regional sensor arrays that are placed in (possibly overlapping) predefined regions. These regions are selected such that the terrain and weather conditions should preferably be statistically homogeneous within that region (e.g., desert land, hilly terrain, and forest area). Each of the regional sensor arrays consists of a number of sensor modules, and the coverage area of each of these modules is possibly non-overlapping. These sensor modules consists of sensors,  $(S_1, S_2, \dots, S_m)$ , of (possibly) different modalities such as passive infrared (PIR), acoustic, and seismic. Each of the sensor modules has a limited processing and communication capability with which it can perform the task of target detection and classification, and communicate the results to a processing center. Let  $c$  be the number of events of interest, that need to be detected and identified; examples include a human or a group of humans ( $C_1$ ), animals led or ridden by humans ( $C_2$ ), and small or large vehicles ( $C_3$ ) crossing the border. Here  $c = 3$  and let  $C_0$  be the event of nominal condition (i.e., no event of interest).

### 2.3. Methodology

Unattended Ground Sensors have to operate in various environmental conditions, and these environmental conditions affect the sensor observations [4]. Hence, the sensor signal for a particular event or for nominal condition, i.e. no event of interest, would change with these external factors. The external factors need not be limited to weather conditions or time of the day, but, it can be any event which affects majority of the sensor observations in a region, for eg: heavy commercial traffic in the morning on highways which affects seismic and acoustic sensor observations. These conditions change with time and also repeat, due to the day-night cycle, annual seasons, etc. So, if the classifiers are trained for a particular condition, they can be used in operational phase, whenever similar conditions are observed. This motivates the need for a model which captures the slow time scale evolution of external factors. In this paper, these external conditions which affect the majority of the sensor observations in a region and which evolve slowly with time are referred to as context. Since, the context affects majority of the

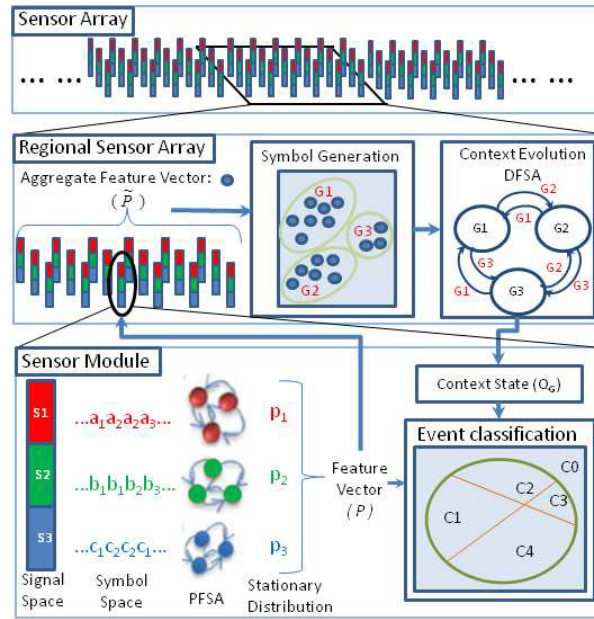


Fig. 1: Schematic representation of the proposed framework

sensors in the region, a sensor array fusion technique is devised to identify it. This process of sensor array fusion and context modeling is explained below.

The time series from each sensor in the array ( $S_1, \dots, S_m$ ) is used to compute the respective stationary state probability row vectors ( $p_1, \dots, p_m$ ), using the tools of SDF [9], which will be described in Section 3.1. These row vectors are concatenated together to make a single feature vector ( $P$ ), which serves as a representation of the multiple time series data from sensors of (possibly) different modality in the sensor module. For every sensor module, there is a single  $P$  vector, implying that there are  $n$  such  $P$  vectors  $P_1, P_2, \dots, P_n$  in a particular region. The context information cannot be inferred from any individual sensor, hence, a feature level sensor fusion is required to extract this information from the time series of sensor arrays. Assuming spatial sparsity of events of interest ( $C_1, \dots, C_c$ ), if the number  $n$  is sufficiently large, the aggregate of the feature vectors  $P_1, P_2, \dots, P_n$  would be minimally affected by the events at few of the sensor modules in the region. This weighted aggregation is performed using features from all the sensors in a region, hence, this is termed as spatial aggregation of features. The spatial aggregate gives us a meta-data feature ( $\tilde{P}$ ) which represents the nominal condition in the region during a particular interval of time. The process of spatial aggregation is performed in the Hilbert space of symbolic systems (HSSS) which was formulated in [10]. The process of aggregation is explained in detail in Section 3.2.

Once this sensor array fusion is carried out, a point in the aggregated feature space is obtained. This process is repeated over time in various conditions to obtain a number of data points. If each point represents a nominal condition of the sensor network, then the points representing similar conditions are clustered together in an unsupervised manner to avoid redundancy. Let, the clusters obtained from the unsupervised clustering be called as  $G_1, G_2, \dots, G_K$ . Here, the number of clusters  $K$  is chosen based on a score which incorporates modeling error and model complexity (see Section 3.3). Once the clusters in the data set are identified, each aggregate feature is labeled with the respective cluster name  $G_1, G_2, \dots, G_K$ . These names are referred to as symbols and the set of these symbols is the input alphabet ( $\Sigma$ ), i.e.,  $\Sigma = \{G_1, G_2, \dots, G_K\}$ . A depth  $D$  is chosen (see section 3.4) and a DFSA model (Defn. 2.1) is constructed using the state transition function ( $\delta$ ) of the D-Markov machine (Remark 2.1). This DFSA model is referred to as the Context Evolution model. Each state of this model is a string of  $D$  symbols, which can be viewed as temporal aggregation of the symbols. Thus, the symbolic representation of the context is essentially spatio-temporal aggregation of features from the sensor array; the state of the DFSA is named as a context state. Whenever, a new aggregated feature vector becomes available during the operational phase, it is assigned to one of the clusters and is labeled by the corresponding cluster name. This cluster label acts

as a symbol for the context evolution model. The DFSA model uses the current context state and the new symbol to generate the new context state which is broadcasted to all sensors in the region.

Traditionally, a classifier is defined as a function mapping a feature space into a finite set. In this paper, the term context-based event classifier is introduced as follows.

**Definition 2.4** (*Context-based Event Classifier*) Let  $Q_G$  be the set of contexts,  $\Psi$  be the set of features, and  $C$  be the (finite) set of event classes. Then, a function  $A : Q_G \times \Psi \rightarrow C$  is said to be a context-based event classifier if  $\forall q \in Q_G, \forall P \in \Psi, A(q, P) \in C$ , i.e.,  $A(q, P)$  is a classifier.

**Remark 2.2** For the case of single context, i.e.,  $|Q_G| = 1$ , the context-based event classifier in Defn. 2.4 degenerates to the classifier  $A : \Psi \rightarrow C$ . Typically the cardinality of  $\Psi$  is uncountable and the function  $A$  is surjective.

It is proposed to first label each feature vector with an event class (from  $C$ ) and a context state (from  $Q_G$ ), then, a classifier is trained for feature vectors with the same context state label to obtain the context-based classifier. The feature vector is generated from time series data at a fast time scale, the aggregation process is at a medium time scale, and the events of interest occur sporadically. Hence, instead of real-time context evaluation, the most recent value of a context state can be used, because a context is assumed to evolve at a slower time scale. In the operational phase, the sensor modules use the most recent broadcasted feedback of context state to choose the classifier and then classify the new feature vector to an event class.

#### 2.4. Dynamic Data Driven Applications Systems (DDDAS)

The proposed framework for context-based event classification is a DDDAS. The context influences the sensor data and introduces dynamics in the system. The framework tracks the aggregate sensor array behavior using the features acquired from all the sensor array nodes to infer the current context. This context is broadcasted as feedback to all sensor nodes to alter the interpretation of the sensor data by choosing appropriate decision boundaries using the context-based classifier (Defn. 2.4). This paper lays the crucial foundation for the unattended ground sensors as a DDDAS to perform context-based target classification. New research is directed to identify regions in feature space  $\Psi$  in different contexts which have inadequate confidence in classification. Using the feedback of context state and knowledge of regions of inadequate confidence, higher fidelity sensors (e.g. Camera mounted on an UAV) would be activated to enhance the sensor suite by adding new sensor data streams, in order to improve classification performance for features in those regions.

### 3. The Underlying Theory

This section presents the theoretical concepts of the proposed methodology in the following four subsections.

#### 3.1. Symbolic Dynamic Filtering for Feature Extraction

This section briefly describes the concepts of Symbolic Dynamic Filtering (SDF) [9] for extracting atomic patterns from single sensor data. While the details on the theory and applications of SDF have been reported in previous publications (e.g., [9][12]), the pertinent concepts are succinctly presented below for completeness.

Symbolic feature extraction from time series data is posed as a two-time-scale problem. The *fast scale* is related to the response time of the process dynamics. Over the span of data acquisition, dynamic behavior of the system is assumed to remain invariant, i.e., the process is quasi-stationary at the fast scale. On the other hand, the *slow scale* is related to the time span over which non-stationary evolution of the system dynamics may occur. It is expected that the features extracted from the fast-scale data will depict statistical changes between two different slow-scale epochs if the underlying system has undergone a change. The method of extracting features from stationary time series data is comprised of the following steps.

- Sensor time series data, generated from a physical system or its dynamical model, are collected at a slow-scale epoch and let it be denoted as  $\mathbf{q}$ . A compact (i.e., closed and bounded) region  $\Omega \in \mathbb{R}^n$ , where  $n \in \mathbb{N}$ , within which the stationary time series is circumscribed, is identified. Let the space of time series data sets be represented as  $Q \subseteq \mathbb{R}^{n \times N}$ , where  $N \in \mathbb{N}$  is sufficiently large for convergence of statistical properties



within a specified threshold. While  $n$  represents the dimensionality of the time-series,  $N$  is the number of data points in the time series. Then,  $\{q\} \in Q$  denotes a time series at the slow-scale epoch of data collection.

- Encoding of  $\Omega$  is accomplished by introducing a partition  $\mathbb{B} = \{B_0, \dots, B_{(|\Sigma|-1)}\}$  consisting of  $|\Sigma|$  mutually exclusive (i.e.,  $B_j \cap B_k = \emptyset \forall j \neq k$ ), and exhaustive (i.e.,  $\bigcup_{j=0}^{|\Sigma|-1} B_j = \Omega$ ) cells, where each cell is labeled by symbols  $\sigma_j \in \Sigma$  and  $\Sigma = \{\sigma_0, \dots, \sigma_{|\Sigma|-1}\}$  is called the alphabet. This process of coarse graining can be executed by uniform, maximum entropy, or any other scheme of partitioning [12]. Then, the time series data points that visit the cell  $B_j, j = 0, 1, \dots, |\Sigma| - 1$  are denoted as  $\sigma_j$ . This step enables transformation of the time series data  $\{q\}$  to a symbol sequence  $\{s\}$ , consisting of the symbols in the alphabet  $\Sigma$ .
- The reference probabilistic finite state automata (PFSA) (see Defn. 2.2) is then constructed from the symbol sequence  $\{s\}$  generated from the training data. Another symbol sequence generated from the test data is run through the PFSA structure to identify the respective probability morph function  $\pi$ . The PFSA considered in this framework is known as D-Markov machine (see Defn. 2.3). Using  $\pi$  and  $\delta$  of this PFSA, the  $(|Q| \times |Q|)$  irreducible state transition probability matrix  $\Pi$  is constructed, where an element  $\Pi_{ij}$  represents the probability of transition from state  $(q_i)$  to state  $(q_j)$ . The left eigenvector of the  $\Pi$  matrix corresponding to the (unique) eigenvalue of 1 is called the stationary state probability vector ( $p$ ) of the PFSA. This  $p$  vector is used as a feature of the time series data.

### 3.2. Aggregation in a Hilbert space of symbolic systems

The symbolic system in the current work is a probabilistic finite state automaton (PFSA) (Defn. 2.2). A stationary probability vector ( $p$ ) can also be represented as a single state PFSA.

In this paper, in order to infer the context from a sensor array, the feature vectors (see Eq. (1)) are aggregated by computing the weighted average from the set of equations (2).

$$P^{(i)} = [p_1^{(i)} p_2^{(i)} \dots p_m^{(i)}] \quad (1)$$

$$\tilde{P} = \begin{bmatrix} \sum_{i=1}^n \alpha_i \otimes p_1^{(i)} \\ \vdots \\ \sum_{i=1}^n \alpha_i \otimes p_m^{(i)} \end{bmatrix}^T \quad (2)$$

In these equation,  $i$  corresponds to each sensor location and  $m$  is the number of co-located sensors.  $\alpha_i$  is a scalar, which is the weight associated with  $i^{th}$  sensor module and  $n$  is the total number of sensor modules in the region. The Eq.(2) shows that the process of aggregating feature vectors  $P$  is equivalent to aggregating the stationary distributions  $p_j \forall j \in \{1, \dots, m\}$  and then concatenating the result. The operation of vector addition and scalar multiplication is not closed on the set of probability vectors, because, apart from being in the finite dimensional Euclidean space, the point is also constrained to lie on a simplex defined by the property that, components of probability vectors sum up to 1. Hence, the Hilbert space of symbolic systems(HSSS) is used, which is closed under the vector addition and scalar multiplication operations.

The vector operations in HSSS are as follows:

1. **Vector addition :** The operation of vector addition on two PFSA has been formulated in [10]. The same operation is performed on the probability vectors in this application. The operation is given below.

$$p_j = [p_{j1}, p_{j2}, \dots, p_{jd}] \forall j \in \{1, 2, 3\}$$

$$p_1 \oplus p_2 = p_3$$

$$p_{3i} = \frac{p_{1i} p_{2i}}{\sum_{j=1}^d p_{1j} p_{2j}} \forall i \in \{1, \dots, d\} \quad (3)$$

2. **Scalar Multiplication :** The operation of scalar multiplication for PFSA has been formulated in [10]. The operation is mentioned below for probability vectors.

$$p_1 \otimes \alpha_1 = p_2$$

$$p_{2i} = \frac{p_{1i}^{\alpha_1}}{\sum_{j=1}^d p_{1j}^{\alpha_1}} \forall i \in \{1, \dots, d\} \quad (4)$$

The weights (i.e.  $\alpha_i \forall i \in \{1, \dots, n\}$ ) are used to provide flexibility to give relative importance to certain areas within a region or certain sensors in the array. Without any information, the weights are uniform and aggregation is just the average. If additional information of few faulty sensors or imprecise sensors is available, then, the weights can be set to zero or decreased respectively.

### 3.3. Unsupervised Clustering

K-means clustering has been used in many unsupervised learning applications. The proposed framework can support any kind of unsupervised clustering algorithm and k-means is just one of the possible techniques. The time series data is de-noised, transformed, symbolized, compressed to the feature vector, and aggregated before clustering, so it becomes difficult to know which conditions would have significantly affected the signals to make the aggregate feature lie in a different cluster. In other words, even if all the conditions are known, which is unlikely, it would be difficult to know *a priori* the number of clusters in the data. However, the number of clusters needs to be mentioned in advance for k-means, so, an Akaike Information Criterion like score is evaluated after clustering the data into different number of clusters ( $K$ ) and the  $K$  with minimum score is used. This Score function has a modeling error term and a model complexity term. Modeling error can be estimated with residual sum of squares (RSS). A parameter  $\lambda$  is chosen by the user to make a tradeoff between the modeling error and model complexity. The general form of this equation is:

$$Score(k) = error(k) + \lambda k \quad (5)$$

Once the number of clusters ( $K$ ) in the dataset is obtained, k-means clustering gives  $K$  disjoint sets of data. The cluster label is assigned to each data point in this set which is then, used as the symbol in the context evolution DFSA as explained in Section 2.3.

### 3.4. Context Modeling

As explained in section 2.3, the context is a spatio-temporal aggregation of features from the sensor network. The context is represented by a state in the context evolution model, which is implemented as a DFSA. Since, the alphabet ( $\Sigma$ ) is obtained from unsupervised clustering, only the parameter depth ( $D$ ) is needed to construct the DFSA model with the D-Markov structure, given by the state transition function in Remark 2.1. The computational complexity, which is dictated by the number of states in the DFSA, grows exponentially with depth of this DFSA model. The training data might be insufficient to train the classifiers for each state of the DFSA model, as some of the states might not be visited very often, if  $D$  is large. These factors prompt using smaller values for depth, but, a larger depth in D-Markov machines provides a longer history resulting in (possibly) better model representation. So, a trade-off has to be made and a suitable depth needs to be chosen according to the available computational resources and desired model accuracy. An alternate way to obtain the context evolution model is to construct a D-Markov Machine (Defn. 2.3) which best fits the symbol sequence using the principles of state-splitting and state-merging, as explained by Patrick *et al.* in [13]. State merging assimilates histories from symbol blocks leading to same symbolic behavior and state splitting splits states by adding more history and hence, increases the depth of the D-Markov machine. The underlying DFSA of this PFSA is then used as the context evolution model.

## 4. Experiments and Results

Previously collected field data [3] were analyzed to validate the proposed concept of context-based classification. The data sets were corrupted by additive Gaussian noise to simulate the effects of different contexts. The test procedure and results are presented in this section.

The seismic, PIR and acoustic sensor data, used in this analysis, were collected on three different days from test fields on a wash (i.e., the dry bed of an intermittent creek) and at a choke point (i.e., a place where the targets are forced to go due to terrain difficulties). The targets consisted of humans (e.g., male and female), animals (e.g., donkeys, mules, and horses), and all-terrain vehicles (ATVs). The humans walked alone and in groups with and without backpacks; the animals were led by their human handlers (simply denoted as animal in this paper) and they made runs with and without payloads; and ATVs moved at different speeds (e.g., 5 mph and 10 mph). There were

Table 1: Variance of additive zero mean Gaussian noise to simulate context

Sensor Modality	Context#1	Context#2	Context#3
Acoustic	0	0.001	0.005
PIR	0	0.01	0.02
Seismic	0	0.002	0.004

three sensor sites, each equipped with seismic, PIR and acoustic sensors. The seismic sensors (geophones) were buried approximately 15 cm deep underneath the soil surface, the PIR and acoustic (microphone) sensors were collocated with the respective seismic sensors. All targets passed by the sensor sites at a distance of approximately 5 m. Signals from both sensors were acquired at a sampling frequency of 10 kHz. Each data set, acquired at a sampling frequency of 10 kHz, has  $10^5$  data points that correspond to 10 seconds of the experimentation time. In order to test the capability of the proposed algorithm for target detection, another data set was collected with no target present. All the samples were collected under similar warm and sunny conditions during the day time; hence, the data is expected to have just one context, as per our context definition.

In order to simulate three types of context, three types of additive zero mean Gaussian noise per sensor modality were imposed on the original time series signals. The Gaussian noise with parameters shown in Table 1 were added to the sensor time series data. Fig. 2 illustrates how the added context affected the time series of seismic sensor and the corresponding feature vectors for a human target.

After the context was added to the data, the next step was to down-sample the data by 100 for computation purposes. It was shown by [3] that information loss by down-sampling this data set by 10 was insignificant. The down-sampling by 100 yielded reasonable classification results while drastically improving computation time. In order to mitigate differences in signal intensities for a given target type due to variations in proximity to the sensor module, the data was made to be zero mean and unit variance. The data was split into two parts: training and testing. The fraction of data kept aside for training is called the training fraction. Training fraction of 70% was chosen, so, 30% of the data was available for testing. This allowed sufficient data to train the classifiers while leaving enough data for testing to properly assess classification performance.

The partitioning of the time series data in the training set is carried out using maximum entropy scheme for each sensor modality. Using SDF (as mentioned in Section 3.1), stationary state probability vectors are extracted from the time series for each sensing modality and they are concatenated together to get a feature vector which represents the event observed by the sensors. The time series observed by each sensor module is randomly chosen from the training data set assuming same context in the full region, using a reasonable probability density over the set of events. This density is referred to as  $PE$ . In Eq. (6),  $pe_i$  represents the probability of  $i^{th}$  event and the set of events  $E = \{No\ event, Human, Animal, Vehicle\}$  enumerates all the events.

$$PE = [pe_0, pe_1, pe_2, pe_3] = [0.90, 0.04, 0.04, 0.02] \quad (6)$$

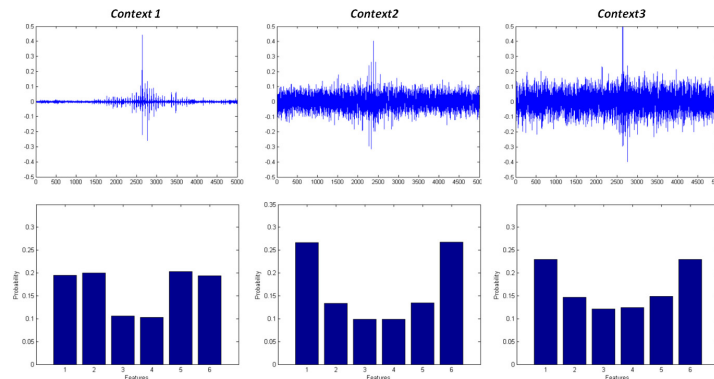


Fig. 2: Effect of context addition on time series data of seismic sensor and feature vectors for a human crossing the sensor site



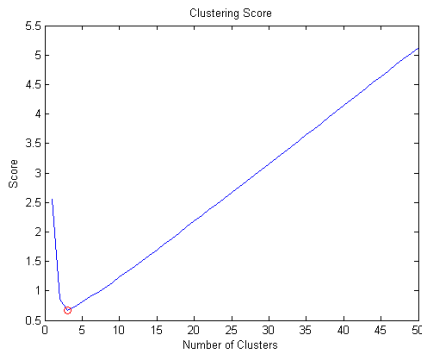


Fig. 3: Clustering Score for different cluster sizes using  $\lambda = 0.1$

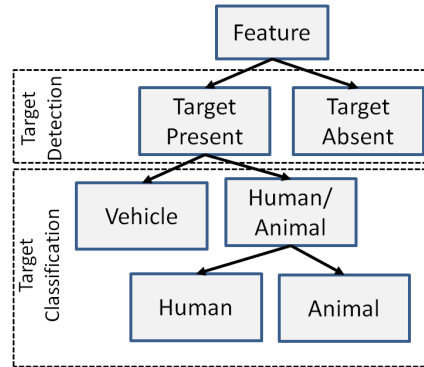


Fig. 4: Binary Classification Tree

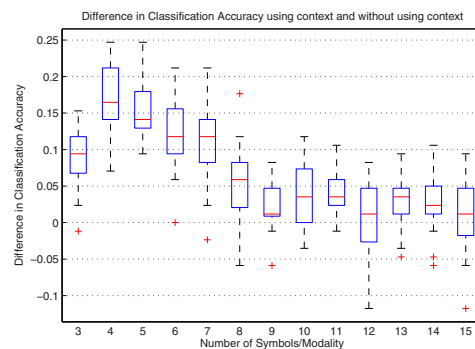
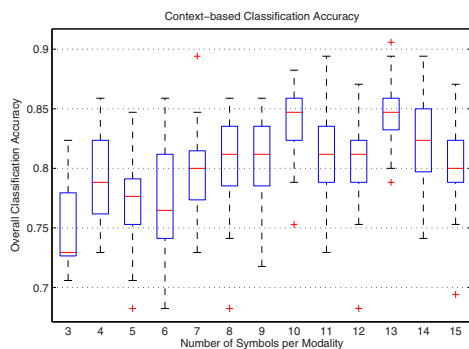


Fig. 5: Comparison using box plots: (a) Context-based classification result for different number of symbols ( $|\Sigma|$ ); (b) Difference in classification accuracy using context and without using context for various number of symbols

After all the sensor modules observed the randomly chosen events at time  $t = 1$ , the process of aggregation of feature vectors, as explained in Section 3.2 was carried out. This process was repeated for  $T$  time steps; here,  $T = 100$ . Unsupervised clustering was performed on all the aggregated feature vectors, as explained in section 3.3. The plot of clustering score v/s number of clusters is shown in Fig. 3, which shows that the minimum score corresponds to number of clusters  $K = 3$ . The context evolution model was constructed as a DFSA with  $|\Sigma| = 3$  and  $D = 1$ . For any given aggregate feature  $\tilde{P}$  used in training, the cluster label and the context state could be inferred from this model. The feature vectors ( $P$ ) corresponding to each  $\tilde{P}$  and the time series corresponding to each  $P$ , which were initially labeled only with events, were then labeled with the context state as well.

The problem of target detection was then formulated as a binary pattern classification, where no target present corresponds to one class, and target present (i.e., human, vehicle or animal) corresponds to the other class. For each context state, a SVM classifier with linear kernel was trained for this target detection task (Level 1 Classification). The process is repeated for Level 2 and Level 3 Classification, after re-evaluating the Maximum Entropy partition for the smaller datasets in which target of interest was present. The binary classification tree is shown in Fig. 4. The partitioning and hence, feature extraction process is repeated for the same alphabet size to provide better class separability between classes of interest at different classification levels. The process of choosing alphabet size is not addressed in this paper, but results for different alphabet sizes are provided.

A comparison of the proposed technique is provided with the traditional method of classification without using the knowledge of context. To make the comparison, 25 different testing and training sets was chosen at random for a given number of symbols while maintaining the 70% training fraction. Overall classification accuracy results were gathered from each of the training-testing pairs and the results are shown in Fig. 5a. Fig. 5b shows a box plot of the difference in classification accuracy using context and without using context as a function of the

alphabet size. This result suggests that context-based classification can be used to improve overall classification performance. The improvement of up to 12% could be shown in the present application for alphabet size of 7 and classification accuracy of 80%. As the number of symbols increase the expressive power of the symbolic system increases and hence, the class separability in the feature space may increase up to a certain point before saturating. If the number of symbols is increased further, the performance actually degrades due to the noise. The classification performance was not optimized in both cases, in order to have uniformity to study only the effect of context. Performance in both cases can be improved by trying out different classifier kernels or classification techniques, by choosing only those sensors which improve class separability and by choosing different alphabet size for each modality. However, the aim of this paper is fulfilled by introducing a new framework for context-based event classification using SDF and successfully verifying that, the knowledge of context helps to perform better than classifiers which do not use context.

## 5. Conclusion and Future Work

This paper presents a dynamic data-driven method of context-based target detection and classification. The proposed method has been tested with field data of unattended ground sensors (UGS) and preliminary test results have been reported. For implementation in practice, the proposed method of target detection and classification needs further theoretical research and validation. To this end, there are many topics of research that must be addressed. As typical examples, three topics of research are presented below.

1. *Novelty detection*: New patterns must be discovered at the aggregated feature level if occurrence of unusual events are consistently observed in the same context.
2. *Decision to include more sensors*: Develop techniques to use prior knowledge and sensor observations to decide if higher fidelity sensors are needed to obtain adequate classification accuracy.
3. *Transfer and supervised learning*: The challenge is to construct classifiers for a new context that was not seen during the training phase. Transfer learning can help to estimate classifiers for new contexts using the knowledge of classifiers in the known contexts. Supervised learning from higher fidelity sensors in a slower time scale would help to adapt the classifier and reduce false alarms.

## Acknowledgement

This material is based upon work supported by, or in part by AFOSR Award No. FA9550-12-1-0270. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsor.

## References

- [1] S. G. Iyengar, P. K. Varshney, T. Damarla, On the detection of footsteps based on acoustic and seismic sensing, in: Forty-First Asilomar Conference on Signals, Systems and Computers ACSSC, 2007, pp. 2248–2252.
- [2] R. E. Bland, Acoustic and seismic signal processing for footstep detection, Master's thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science (2006).
- [3] X. Jin, S. Sarkar, A. Ray, S. Gupta, T. Damarla, Target detection and classification using seismic and pir sensors, IEEE Sensors Journal 12 (6) (2012) 1709–1718.
- [4] J. McKenna, M. McKenna, Effects of local meteorological variability on surface and subsurface seismic-acoustic signals, in: 25th Army Science Conference, 2006.
- [5] D. Wilson, D. Marlin, S. Mackay, Acoustic/seismic signal propagation and sensor performance modeling, in: SPIE, Vol. 6562, 2007.
- [6] G. Toussaint, The use of context in pattern recognition, Pattern Recognition 10 (1978) 189–204.
- [7] E. Elnahrawy, B. Nath, Context-aware sensors, in: EWSN, 2004.
- [8] J. Zhao, R. Govindan, D. Estrin, Computing aggregates for monitoring wireless sensor networks, in: IEEE ICC Workshop on Sensor Network Protocols and Applications, Anchorage, AK, USA, 2003.
- [9] A. Ray, Symbolic dynamic analysis of complex systems for anomaly detection, Signal Processing 84 (7) (2004) 1115–1130.
- [10] Y. Wen, A. Ray, Vector space formulation of probabilistic finite state automata, Journal of Computer and System Sciences 78 (2012) 1127–1141.
- [11] X. Jin, S. Gupta, K. Mukherjee, A. Ray, Wavelet-based feature extraction using probabilistic finite state automata for pattern classification, Pattern Recognition 44 (7) (2011) 1343–1356.
- [12] V. Rajagopalan, A. Ray, Symbolic time series analysis via wavelet-based partitioning, Signal Processing 86 (11) (2006) 3309–3320.
- [13] P. Adenis, K. Mukherjee, A. Ray, State splitting and state merging in probabilistic finite state automata, in: American Control Conference, 2011.